UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/715,459	11/19/2003	Curtis Davis	NVDA/AG-08-0112-US	3916

26290 7590 06/28/2010

PATTERSON & SHERIDAN, L.L.P.
3040 POST OAK BOULEVARD
SUITE 1500
HOUSTON, TX 77056

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
06/28/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/715,459	DAVIS ET AL.	
	Examiner	Art Unit	
	KEITH VICARY	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 May 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7,10-17,20-23,25-36 and 38-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7,10-17,20-23,25-36 and 38-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :3/3/2010, 3/15/2010, 5/17/2010.

DETAILED ACTION

1. Claims 1-5, 7, 10-17, 20-23, 25-36, and 38-40 are pending in this office action and presented for examination. Claims 1, 25, and 31 are newly amended and claims 24 and 37 are newly cancelled by amendment filed 5/5/2010.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-5, 7, 10-17, 20-23, 25-36, and 38-40 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

4. Claim 1 recites the limitation "allocating memory resources within the PPM to the DME and the FPE" in lines 11-12. However, the original disclosure does not appear to support this limitation. For example, page 18 of the original specification discloses that "[the PCE] manages the definition and allocation of all internal and external memories". However, the aforementioned citation, as well as what appears to be the specification as a whole, does not appear to implicitly or explicitly disclose that the recited allocation is "to the DME" and "to the FPE".

a. Claims 2-5, 7, 10-17, 20-23, and 38 are rejected for failing to alleviate the rejection of claim 1 above.

5. Claim 25 recites the limitation "allocating memory resources within the PPM to the DME and the FPE" in lines 12-13. However, the original disclosure does not appear to support this limitation. For example, page 18 of the original specification discloses that "[the PCE] manages the definition and allocation of all internal and external memories". However, the aforementioned citation, as well as what appears to be the specification as a whole, does not appear to implicitly or explicitly disclose that the recited allocation is "to the DME" and "to the FPE".

b. Claims 26-30 and 39 are rejected for failing to alleviate the rejection of claim 25 above.

6. Claim 31 recites the limitation "allocating memory resources within the PPM to the DME and the FPE" in lines 12-13. However, the original disclosure does not appear to support this limitation. For example, page 18 of the original specification discloses that "[the PCE] manages the definition and allocation of all internal and external memories". However, the aforementioned citation, as well as what appears to be the specification as a whole, does not appear to implicitly or explicitly disclose that the recited allocation is "to the DME" and "to the FPE".

c. Claims 32-36 and 40 are rejected for failing to alleviate the rejection of claim 31 above.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1-5, 7, 10-14, 16, 20-23, 25, and 29-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulick et al. (Gulick) (US 5692211) in view of Bishop et al. (Bishop) (SPARTA: Simulation of Physics on a Real-Time Architecture) in view of Mohamed (US 6366998) in view of Dixon et al. (Dixon) (US 6754732).

9. Consider claim 1, Gulick discloses a Central Processing Unit (CPU) (Figure 1, CPU 102; explained in col. 4, line 64) coupled to an external memory (Figure 1, hard disk 122) and one or more peripherals (for example, Figure 1, network interface controller 124); and a multimedia processing unit (MPU) coupled to the CPU (Figure 1, multimedia engine 112) and a Multimedia Processing Memory (MPM) (Figure 1, main memory 110) and comprising a MPU Control Engine (MCE) (Figure 1, multimedia engine 112), a Data Movement Engine (DME) (Figure 3, DMA transfer engine 164), a DME control interface (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main

memory 110 to the multimedia memory 160) and a Multimedia Processing Engine (MPE) (Figure 3, DSP Engine 210), wherein the MCE is configured to control the overall operation of the MPU by receiving multimedia simulation requests from the CPU (col. 7, lines 14-25, the multimedia engine includes command buffers which store commands received from the CPU) via a MPU software driver executing on the CPU (col. 9, lines 56-59, disclose of a CPU executing driver software to generate multimedia data), issuing commands to the DME (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and the MPE to perform multimedia simulation computations associated with the multimedia simulation requests (col. 7, lines 23-25, multimedia instructions from the CPU which are executed by the DSP engine 210 on multimedia data), and allocating memory resources within the MPM to the DME and the MPE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160; in other words, the PCE allocates particular memory resources in the main memory 110 to the DME and the MPE by transferring commands to the DME which causes the DME to access those memory resources and transfer the contents stored within for the MPE to use in processing), the DME is configured to transfer multimedia data between the MPM and at least one MPU internal memory (Figure 1, multimedia memory 160) in response to commands received from the MCE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine

in transferring data from the main memory 110 to the multimedia memory 160) and to initiate context switches relative to one or more banks of the at least one MPU internal memory (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer), the DME control interface is coupled to each of the MCE and the DME (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160), and the MPE is configured to perform operations to generate multimedia simulation data associated with the multimedia simulation requests, wherein each operation is issued to the MPE by the MCE (col. 7, lines 23-25, the command buffers 16 may also receive multimedia instructions from the CPU 102 which are executed by the DSP engine 210 on multimedia data; col. 7, lines 65-67, DSP engine 210 provides digital video pixel data).

However, Gulick does not disclose that the system processes physics as opposed to or in addition to multimedia (and thus does not disclose of, for example, a *physics* processing unit, a *physics* processing memory, *physics* simulation requests, *physics* data and so forth), and does not explicitly disclose that the DSP engine performs floating point operations (and thus does not disclose of a *floating point*

engine). Gulick also does not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick also does not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Bishop discloses of offloading physics calculations to specialized hardware (Figure 1, ideal SPARTA implementation; section 1 discloses of physical modeling of solid objects through the use of specialized hardware; section 2 and section 3.1 discloses of collision detection and force computation) and discloses of that specialized hardware performing floating point operations (for example, page 5, which discloses of flexible floating point hardware in bullet point 5 and FLOP performance).

Bishop's teaching of using specialized hardware accelerates physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments (Bishop, section 1).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bishop with the invention of Gulick in order to accelerate physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments. Note that Gulick's teaching of using specialized hardware for physics simulation, when applied to the invention of Gulick,

teaches that the Multimedia Processing Memory (PM) is a Physics Processing Memory (PPM), the MPU Control Engine (MCE) is a PPU Control Engine (PCE), the Multimedia Processing Engine (MPE) is a Floating Point Engine (FPE), the multimedia simulation computations are physics simulation computations, the multimedia simulation requests are physics simulation requests, the multimedia data is physics data, and the multimedia operations are floating point operations.

However, Gulick and Bishop do not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick and Bishop also do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Mohamed discloses of a floating point engine (col. 8, lines 47-48 disclose that the functional units described therein can be floating point units) that is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW) (col. 5, lines 4-8, disclose of a VLIW instruction packet which can simultaneously utilize different functional units).

Mohamed teaches that data parallelism models are very efficient in block based applications (Mohamed, col. 2, lines 1-7).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Mohamed with the invention of Gulick and Bishop in order to improve efficiency in block based applications.

However, Gulick, Bishop, and Mohamed do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Dixon discloses of a first packet queue for receiving command packets from a control engine and transmitting the command packets to a DME (col. 3, lines 39-46, descriptors are generally queued in a descriptor queue or command queue, and the DMA then services each descriptor) and a second packet queue receiving response packets from the DME and transmitting the response packets to the CE (col. 4, lines 15-17, oftentimes a device may have associated with it a buffer or queue for the actual data that it transmits to another location and/or receives from another location).

Dixon's teaching supports the situation in which a DMA controller receives more data transfer requests than it can service at one time (Dixon, col. 3, lines 39-41), and it would be readily recognized to one of ordinary skill in the art at the time of the invention that a buffer for receiving data is used in a likewise manner, to support the situation in which data is received faster than it can be used.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dixon with the invention of Gulick, Bishop, and Mohamed in order to support the situation in which data transfer requests and data are received faster than they can be serviced.

10. Consider claim 2, Gulick discloses the CPU comprises a processing unit resident in a personal computer (col. 1, lines 33-34, personal computer systems).

11. Consider claim 3, Gulick discloses the CPU comprises a processing unit resident in a game console (col. 1, lines 33-34, personal computer systems; it would be readily recognized to one of ordinary skill in the art at the time of the invention that personal computers can execute games).

12. Consider claim 4, Gulick discloses a Graphics Processing Unit (GPU) operatively connected to the CPU (col. 2, line 1, video accelerator cards).

13. Consider claim 5, Gulick discloses the CPU and PPU communicate via at least one selected from a group of physical interfaces consisting of: Universal Serial Bus (USB), USB2, Firewire, Peripheral Component Interconnect (PCI), Peripheral Component Interconnect Extended (PCI-X), PCI-Express, and Ethernet (col. 6, lines 8-11, the CPU provides commands through the PCI bus 120 to the multimedia engine 112).

14. Consider claim 7, Gulick discloses the PCE comprises programming code stored in a memory resident within the PPU (Figure 3, multimedia memory 160, in conjunction with col. 7, line 12, commands; alternatively, col. 7, lines 60-61, the multimedia engine includes firmware).

15. Consider claim 10, Gulick discloses the PPM comprises high-speed memory (col. 5, line 7, DRAM) and the PPU further comprises a high-speed data bus connecting the high-speed memory to at least one of the DME and the FPE (Figure 1, line connecting main memory 110 and multimedia engine 112; Figure 3 shows the connection to the DSP engine and the DMA transfer engine).

16. Consider claim 11, Gulick discloses a memory interface unit managing data communication between the high-speed data bus and the high-speed memory (Figure 1, main memory 110 and line connecting main memory 110 to multimedia engine 112; the input/output pins and address pins of the main memory manage data communication between the memory cells of the memory and the data bus).

17. Consider claim 12, Gulick discloses a processor bus connecting the PCE with at least one physical interface to the CPU (Figure 1, line connecting the command buffers 166 with the chip set 106).

18. Consider claim 13, Gulick discloses the processor bus is separate from the high-speed bus and connected to the high speed bus via a bridge (Figure 1 shows the line connecting the command buffers 166 with the chip set 106, and the line connecting main memory 110 and multimedia engine 112, to be separate; the two are bridged by at least the DMA transfer engine).

19. Consider claim 14, Gulick discloses an Inter-Engine Memory (IEM) coupled to the DME and the FPE (Figure 1, multimedia memory 160, shown coupled to the DSP engine 210 and DMA transfer engine 164) and configured to receive physics simulation data from the PPM in response to commands received from the DME (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands received by the DMA engine for transferring data from the main memory 110 to the multimedia memory 160).

20. Consider claim 16, Gulick discloses the IEM comprises multiple banks of memory adapted to support parallel threads of execution (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer).

21. Consider claim 20, Gulick discloses a Scratch Pad Memory (SPM) configured to receive data from the PPM in response to commands from the DME (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160); wherein the IEM further comprises a first bank accessible to the DME and a second bank accessible to the FPE (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer); and, wherein the DME further comprises: a first unidirectional crossbar connected to the first bank (the read address line from the DME to the main memory which tells the main memory where data to be sent to the multimedia memory is located; this is connected to the first bank by the multimedia memory, which sends the data to the first bank); a second unidirectional crossbar connected to the second bank (the write address line from the DME to the multimedia memory which tells the multimedia memory where data from the main memory will be written, of which the second bank is a part of); and, a bi-directional crossbar connecting the first and second crossbars to at least one of the PPM or SPM (col. 6, lines 13-16, disclose of arbitration logic which determines access to the main memory; in other words, both the CPU and the multimedia engine access the main memory and thus there must be connections from the main memory to both the CPU

and the multimedia engine, which is the crossbar, and is collectively bi-directional in that the main memory can be both read and written).

22. Consider claim 21, Gulick discloses a first Address Generation Unit providing Read address data to the first unidirectional crossbar (Figure 3, the portion of the DMA transfer engine which sends the read address to the main memory which tells the main memory where data to be sent to the multimedia memory is located); and, a second Address Generation Unit providing Write address data to the second unidirectional crossbar (Figure 3, the portion of the DMA transfer engine which sends the write address from the DME to the multimedia memory which tells the multimedia memory where data from the main memory will be written).

23. Consider claim 22, Mohamed discloses of a plurality of floating point operation execution units (col. 8, lines 47-48 disclose that the functional units can be floating point units; col. 5, lines 9-17, for example, disclose of using multiple functional units to execute a vector instruction).

24. Consider claim 23, Mohamed discloses the plurality of floating point execution units are selectively grouped together to form a vector floating point unit (col. 5, lines 9-17, for example, disclose that a programmer may use multiple functional units to execute a vector instruction; note the alternate selection of col. 5, lines 1-8, of using the functional units separately to execute each instruction of a VLIW instruction packet).

25. Consider claim 25, Gulick discloses a host, wherein the host comprises an external memory (Figure 1, hard disk 122) and a peripheral (for example, Figure 1, network interface controller 124) coupled to a Central Processing Unit (CPU) (Figure 1, CPU 102; explained in col. 4, line 64); and, a multimedia processing unit (MPU) coupled to the CPU (Figure 1, multimedia engine 112) and a Multimedia Processing Memory (MPM) (Figure 1, main memory 110) and comprising a MPU Control Engine (MCE) (Figure 1, multimedia engine 112), a Data Movement Engine (DME) (Figure 3, DMA transfer engine 164), a DME control interface (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and a Multimedia Processing Engine (MPE) (Figure 3, DSP Engine 210), wherein the MCE is configured to control the overall operation of the MPU by receiving multimedia simulation requests from the CPU (col. 7, lines 14-25, the multimedia engine includes command buffers which store commands received from the CPU) via a MPU software driver executing on the CPU (col. 9, lines 56-59, disclose of a CPU executing driver software to generate multimedia data) issuing commands to the DME (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and the MPE to perform multimedia simulation

computations associated with the multimedia simulation requests (col. 7, lines 23-25, multimedia instructions from the CPU which are executed by the DSP engine 210 on multimedia data), and allocating memory resources within the MPM to the DME and the MPE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160; in other words, the PCE allocates particular memory resources in the main memory 110 to the DME and the MPE by transferring commands to the DME which causes the DME to access those memory resources and transfer the contents stored within for the MPE to use in processing), the DME is configured to transfer multimedia data between the MPM and at least one MPU internal memory (Figure 1, multimedia memory 160) in response to commands received from the MCE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and to initiate context switches relative to one or more banks of the at least one MPU internal memory (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer), the DME control interface is coupled to each of the MCE and the DME (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store

commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160), and the MPE is configured to perform operations to generate multimedia simulation data associated with the multimedia simulation requests, wherein each operation is issued to the MPE by the MCE (col. 7, lines 23-25, the command buffers 16 may also receive multimedia instructions from the CPU 102 which are executed by the DSP engine 210 on multimedia data; col. 7, lines 65-67, DSP engine 210 provides digital video pixel data); wherein the host stores a main program (col. 9, line 65, application executing on the CPU) and the MPU software driver (col. 9, lines 56-59, disclose of a CPU executing driver software to generate multimedia data); and, wherein the MPU software driver manages all communication between the MPU and the CPU (col. 9, lines 56-59, disclose of a CPU executing driver software to generate multimedia data; col. 11, lines 34-37, also discloses of the driver software writing the high level commands).

However, Gulick does not disclose that the main program is a game program, and that the system processes physics as opposed to or in addition to multimedia (and thus does not disclose of, for example, a *physics* processing unit, a *physics* processing memory, *physics* simulation requests, *physics* data and so forth), and does not explicitly disclose that the DSP engine performs floating point operations (and thus does not disclose of a *floating point* engine). Gulick also does not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick also does not disclose that the DME control

interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, the examiner takes official notice that personal computers can store and execute games.

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that a personal computer storing and executing games increases the capability of the personal computer and is desirable from an amusement standpoint.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention for the personal computer of Gulick to store and execute game programs.

However, Gulick (as modified by the gaming limitation above) does not disclose that the main program is a game program, and that the system processes physics as opposed to or in addition to multimedia (and thus does not disclose of, for example, a *physics* processing unit, a *physics* processing memory, *physics* simulation requests, *physics* data and so forth), and does not explicitly disclose that the DSP engine performs floating point operations (and thus does not disclose of a *floating point* engine). Gulick also does not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick also does not disclose that the DME control interface comprises a first packet queue for

receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Bishop discloses of offloading physics calculations to specialized hardware (Figure 1, ideal SPARTA implementation; section 1 discloses of physical modeling of solid objects through the use of specialized hardware; section 2 and section 3.1 discloses of collision detection and force computation) and discloses of that specialized hardware performing floating point operations (for example, page 5, which discloses of flexible floating point hardware in bullet point 5 and FLOP performance).

Bishop's teaching of using specialized hardware accelerates physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments (Bishop, section 1).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bishop with the invention of Gulick in order to accelerate physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments. Note that Gulick's teaching of using specialized hardware for physics simulation, when applied to the invention of Gulick, teaches that the Multimedia Processing Memory (PM) is a Physics Processing Memory (PPM), the MPU Control Engine (MCE) is a PPU Control Engine (PCE), the Multimedia Processing Engine (MPE) is a Floating Point Engine (FPE), the multimedia simulation computations are physics simulation computations, the multimedia simulation requests

are physics simulation requests, the multimedia data is physics data, and the multimedia operations are floating point operations.

However, Gulick and Bishop do not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick and Bishop also do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Mohamed discloses of a floating point engine (col. 8, lines 47-48 disclose that the functional units described therein can be floating point units) that is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW) (col. 5, lines 4-8, disclose of a VLIW instruction packet which can simultaneously utilize different functional units).

Mohamed teaches that data parallelism models are very efficient in block based applications (Mohamed, col. 2, lines 1-7).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Mohamed with the invention of Gulick and Bishop in order to improve efficiency in block based applications.

However, Gulick, Bishop, and Mohamed do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE

and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Dixon discloses of a first packet queue for receiving command packets from a control engine and transmitting the command packets to a DME (col. 3, lines 39-46, descriptors are generally queued in a descriptor queue or command queue, and the DMA then services each descriptor) and a second packet queue receiving response packets from the DME and transmitting the response packets to the CE (col. 4, lines 15-17, oftentimes a device may have associated with it a buffer or queue for the actual data that it transmits to another location and/or receives from another location).

Dixon's teaching supports the situation in which a DMA controller receives more data transfer requests than it can service at one time (Dixon, col. 3, lines 39-41), and it would be readily recognized to one of ordinary skill in the art at the time of the invention that a buffer for receiving data is used in a likewise manner, to support the situation in which data is received faster than it can be used.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dixon with the invention of Gulick, Bishop, and Mohamed in order to support the situation in which data transfer requests and data are received faster than they can be serviced.

26. Consider claim 29, Mohamed discloses of a dedicated vector processor adapted to perform parallel floating point operations (col. 8, lines 47-48 disclose that the functional units can be floating point units; col. 5, lines 9-17, for example, disclose of using multiple functional units to execute a vector instruction).

27. Consider claim 30, Gulick discloses the PPU further comprises a high-speed memory (col. 8, lines 58-60, the multimedia memory preferable comprises high speed VRAM or DRAM).

28. Consider claim 31, Gulick discloses a multimedia processing unit (MPU) (Figure 1, multimedia engine 112) coupled to a Multimedia Processing Memory (MPM) (Figure 1, main memory 110) and comprising a MPU Control Engine (MCE) (Figure 1, multimedia engine 112), a Data Movement Engine (DME) (Figure 3, DMA transfer engine 164), a DME control interface (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and a Multimedia Processing Engine (MPE) (Figure 3, DSP Engine 210), wherein the MCE is configured to control the overall operations of the MPU by receiving multimedia simulation requests from a Central Processing Unit (CPU) (Figure 1, CPU 102; explained in col. 4, line 64; col. 7, lines 14-25, the multimedia engine includes command buffers which store

commands received from the CPU) via a MPU software driver executing on the CPU (col. 9, lines 56-59, disclose of a CPU executing driver software to generate multimedia data) issuing commands to the DME (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and the MPE to perform multimedia simulation computations associated with the multimedia simulation requests (col. 7, lines 23-25, multimedia instructions from the CPU which are executed by the DSP engine 210 on multimedia data), and allocating memory resources within the MPM to the DME and the MPE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160; in other words, the PCE allocates particular memory resources in the main memory 110 to the DME and the MPE by transferring commands to the DME which causes the DME to access those memory resources and transfer the contents stored within for the MPE to use in processing), the DME is configured to transfer physics data between the MPM and at least one MPU internal memory (Figure 1, multimedia memory 160) in response to commands received from the MCE (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160) and to initiate context switches relative to one or more banks of the at least one MPU internal memory (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main

memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer), the DME control interface is coupled to each of the MCE and the DME (interconnection connecting the DMA transfer engine with the command buffer circuitry and multimedia memory 160, as is necessary in view of col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands for use by the DMA engine in transferring data from the main memory 110 to the multimedia memory 160), and the MPE is configured to perform operations to generate multimedia simulation data associated with the multimedia simulation requests, wherein each operation is issued to the MPE by the MCE (col. 7, lines 23-25, the command buffers 16 may also receive multimedia instructions from the CPU 102 which are executed by the DSP engine 210 on multimedia data; col. 7, lines 65-67, DSP engine 210 provides digital video pixel data).

However, Gulick does not disclose that the system processes physics as opposed to or in addition to multimedia (and thus does not disclose of, for example, a *physics* processing unit, a *physics* processing memory, *physics* simulation requests, *physics* data and so forth), and does not explicitly disclose that the DSP engine performs floating point operations (and thus does not disclose of a *floating point* engine). Gulick also does not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick also does not disclose that the DME control interface comprises a first packet queue for

receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Bishop discloses of offloading physics calculations to specialized hardware (Figure 1, ideal SPARTA implementation; section 1 discloses of physical modeling of solid objects through the use of specialized hardware; section 2 and section 3.1 discloses of collision detection and force computation) and discloses of that specialized hardware performing floating point operations (for example, page 5, which discloses of flexible floating point hardware in bullet point 5 and FLOP performance).

Bishop's teaching of using specialized hardware accelerates physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments (Bishop, section 1).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Bishop with the invention of Gulick in order to accelerate physical models of specialized hardware to achieve real-time performance for creating realistic virtual environments. Note that Gulick's teaching of using specialized hardware for physics simulation, when applied to the invention of Gulick, teaches that the Multimedia Processing Memory (PM) is a Physics Processing Memory (PPM), the MPU Control Engine (MCE) is a PPU Control Engine (PCE), the Multimedia Processing Engine (MPE) is a Floating Point Engine (FPE), the multimedia simulation computations are physics simulation computations, the multimedia simulation requests

are physics simulation requests, the multimedia data is physics data, and the multimedia operations are floating point operations.

However, Gulick and Bishop do not disclose that the aforementioned floating point engine is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW). Gulick and Bishop also do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Mohamed discloses of a floating point engine (col. 8, lines 47-48 disclose that the functional units described therein can be floating point units) that is configured to perform multiple, parallel floating point operations, wherein each parallel floating point operation is specified by a very long instruction word (VLIW) (col. 5, lines 4-8, disclose of a VLIW instruction packet which can simultaneously utilize different functional units).

Mohamed teaches that data parallelism models are very efficient in block based applications (Mohamed, col. 2, lines 1-7).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Mohamed with the invention of Gulick and Bishop in order to improve efficiency in block based applications.

However, Gulick, Bishop, and Mohamed do not disclose that the DME control interface comprises a first packet queue for receiving command packets from the PCE

and transmitting the command packets to the DME, and a second packet queue receiving response packets from the DME and transmitting the response packets to the PCE.

On the other hand, Dixon discloses of a first packet queue for receiving command packets from a control engine and transmitting the command packets to a DME (col. 3, lines 39-46, descriptors are generally queued in a descriptor queue or command queue, and the DMA then services each descriptor) and a second packet queue receiving response packets from the DME and transmitting the response packets to the CE (col. 4, lines 15-17, oftentimes a device may have associated with it a buffer or queue for the actual data that it transmits to another location and/or receives from another location).

Dixon's teaching supports the situation in which a DMA controller receives more data transfer requests than it can service at one time (Dixon, col. 3, lines 39-41), and it would be readily recognized to one of ordinary skill in the art at the time of the invention that a buffer for receiving data is used in a likewise manner, to support the situation in which data is received faster than it can be used.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Dixon with the invention of Gulick, Bishop, and Mohamed in order to support the situation in which data transfer requests and data are received faster than they can be serviced.

29. Consider claim 32, Gulick or alternatively Bishop discloses the PPU is operatively connected within the PC by means of an expansion board (Gulick, col. 1, line 60, multimedia hardware card; alternatively, Bishop, Figure 1, wherein the SPARTA chip interfaces to the host CPU via the control logic).

30. Consider claim 33, Gulick discloses a Graphics Processing Unit (GPU) adapted to compute graphics data for incorporation within execution of a program (Gulick, col. 2, line 1, video accelerator card). However, Gulick, Bishop, Mohamed, and Dixon do not explicitly disclose that the program being executed by the host (col. 9, line 65, application executing on the CPU) is a game program.

On the other hand, the examiner takes official notice that personal computers can store and execute games.

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that a personal computer storing and executing games increases the capability of the personal computer and is desirable from an amusement standpoint.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention for the combination of Gulick, Bishop, Mohamed, and Dixon to store and execute game programs.

31. Consider claim 34, Gulick discloses the general purpose microprocessor generates a command in response to execution of a program and communicates the command to the PPU (Figure 1, CPU 202; explained in col. 4, line 64; col. 7, lines 14-

25, the multimedia engine includes command buffers which store commands received from the CPU; col. 9, line 65, application executing on the CPU).

However, Gulick, Bishop, Mohamed, and Dixon do not explicitly disclose that the program is a game program.

On the other hand, the examiner takes official notice that personal computers can store and execute games.

It would have been readily recognized to one of ordinary skill in the art at the time of the invention that a personal computer storing and executing games increases the capability of the personal computer and is desirable from an amusement standpoint.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention for the combination of Gulick, Bishop, Mohamed, and Dixon to store and execute game programs.

32. Consider claim 35, Gulick discloses the CPU and PPU communicate via at least one selected from a group of physical interfaces consisting of: Universal Serial Bus (USB), USB2, Firewire, Peripheral Component Interconnect (PCI), Peripheral Component Interconnect Extended (PCI-X), PCI-Express, and Ethernet (col. 6, lines 8-11, the CPU provides commands through the PCI bus 120 to the multimedia engine 112).

33. Consider claim 36, Mohamed discloses of a vector processor adapted to run parallel floating point operations (col. 8, lines 47-48 disclose that the functional units can

be floating point units; col. 5, lines 9-17, for example, disclose of using multiple functional units to execute a vector instruction).

34. Claims 15 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulick, Bishop, Mohamed, and Dixon as applied to claim 14 above, and further in view of Humphrey et al. (Humphrey) (US 4933846).

35. Consider claim 15, Gulick discloses DME operation in response to a PCE command (col. 7, lines 14-18, command buffers which store commands...these commands comprise transfer commands received by the DMA engine for transferring data from the main memory 110 to the multimedia memory 160); however, Gulick, Bishop, Mohamed, and Dixon do not explicitly disclose of an Inter-Engine Register (IER) coupled to the DME and the FPE and adapted to initiate DME operation.

On the other hand, Humphrey discloses of an Inter-Engine Register (IER) coupled to a DMA engine and an execution unit and adapted to initiate DMA engine operation (col. 22, lines 5-10, discloses of double-buffering for the DMA controller such that the controlling processor can update the inactive descriptor while maintaining uninterrupted DMA transfer; the IER is described in col. 22, lines 48-53, namely the registers that contain the range and length of each address sequence specified to the DMA controller which correlate to the descriptor; DMA accesses need these addresses which determine the location of the relevant data in order to be initiated; note the attached device of col. 21, line 59, correlates to the execution unit).

Humphrey's teaching of updating an inactive descriptor while maintaining uninterrupted DMA transfer (Humphrey, col. 22, lines 5-9) would be readily recognized to one of ordinary skill in the art at the time of the invention to increase system performance.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Humphrey with the invention of Gulick, Bishop, Mohamed, and Dixon in order to increase system performance.

36. Consider claim 17, Humphrey discloses the IER comprises multiple banks of registers (col. 22, lines 5-10, discloses of double-buffering for the DMA controller such that the controlling processor can update the inactive descriptor while maintaining uninterrupted DMA transfer; the IER is described in col. 22, lines 48-53, namely the registers that contain the range and length of each address sequence specified to the DMA controller which correlate to the descriptor), and Gulick discloses the IEM comprises multiple banks of memory adapted to support two parallel threads of execution (col. 7, lines 7-13, the multimedia memory 160 is partitioned into two or more separate address spaces or buffers, and the DMA engine 164 transfers data from main memory 110 to a first address space or buffer in the multimedia memory 160 while the DSP engine 210 accesses commands and data from the other address space or buffer).

37. Claims 26-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulick, Bishop, Mohamed, and Dixon as applied to claim 25 above, and further in view of Telekinesys (Havok Game Dynamics SDK).

38. Consider claim 26, Gulick, Bishop, Mohamed, and Dixon do not disclose a first Application Programming Interface (API) associated with the main game program; and a second API associated with the PPU driver.

However, Telekinesys discloses of a first Application Programming Interface (API) associated with the main game program (page 18, first paragraph, the toolkit layer); and a second API associated with the PPU driver (page 18, first paragraph, core physics API).

The API for Telekinesys allows easy access to features without needing to dig deep into the workings of the core (Telekinesys, page 18, first paragraph), although examiner notes that APIs in general promote ease of use in this manner.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Telekinesys with the invention of Gulick, Bishop, Mohamed, and Dixon in order to result in ease of use.

39. Consider claim 27, Telekinesys discloses the second API is callable by the first API (page 18, see, for example, the description cell in the toolkit functions feature row; calling one of the toolkit functions would itself call the core physics API as explained in the first paragraph of page 18).

40. Consider claim 28, Gulick and Telekinesys disclose the host further comprises a Graphics Processor Unit (GPU) (Gulick, col. 2, line 1, video accelerator card), wherein the host further stores: a GPU driver and a third API associated with the GPU driver (Telekinesys, benefit cell under the DirectX8 Display library row; industry standard graphics API); wherein the second API is callable by the first (Telekinesys, page 18, see, for example, the description cell in the toolkit functions feature row; calling one of the toolkit functions would itself call the core physics API as explained in the first paragraph of page 18) and third APIs (Telekinesys, page 19, first paragraph, interface the physics system to the existing 3D graphics/rendering solution).

41. Claims 38-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gulick, Bishop, Mohamed, and Dixon as applied to claims 1, 25, and 31 above, and further in view of Hirahara et al. (Hirahara) (US 5063498).

42. Consider claims 38-40, Gulick discloses the PCE is configured to call microcode routines executed within the DME and FPE to perform physics simulation computations (col. 7, lines 33-35, microcode corresponding to video and audio processing instructions or commands).

However, Gulick, Bishop, Mohamed, and Dixon do not explicitly disclose that the DME executes microcode routines (i.e. in processing the transfer instructions).

On the other hand, Hirahara discloses of a DMA controller executing microcode routines (for example, col. 2, lines 12-13, micro-codes to achieve the direct memory access operation).

Hirahara's teaching realizes high speed DMA data transfer with a relatively small hardware increase (Hirahara, col. 2, lines 1-2).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Hirahara with the invention of Gulick, Bishop, Mohamed, and Dixon in order to realize high speed DMA data transfer with a relatively small hardware increase.

Response to Arguments

43. Applicant notes that the amended claim 1 recites the limitations of a floating point engine that includes a vector processor configured to perform multiple, parallel floating point operations to generate physics simulation data associated with physical simulation requests, where each parallel floating point operation is specified by a very long instruction word (VLIW) that is issued to the FPE by a PPU control engine (PCE). Applicant argues that Gulick does not disclose these limitations. Applicant further argues that the remaining references fail to cure these deficiencies of Gulick.

However, the Mohamed reference, previously used to teach the VLIW related limitations in the relevant dependent claims, is just as applicable to the limitation recited above, such that the Mohamed reference applied to the prior art combination used to reject claim 1 of the previous office action would teach the amended claim 1. As with the previous rejection, Gulick teaches of sending a command to a multimedia processing engine and Bishop is used to motivate that command being a floating point command used to perform physics calculations. Mohamed is used in an analogous

manner as it had been used to teach the relevant dependent claims, namely to teach and motivate that floating point command to be a VLIW command for performance and efficiency purposes (e.g. executing multiple floating point operations simultaneously results in greater performance and efficiency than executing floating point operations one at a time).

Conclusion

44. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

45. Any inquiry concerning this communication or earlier communications from the examiner should be directed to KEITH VICARY whose telephone number is (571)270-

1314. The examiner can normally be reached on Monday - Thursday, 7:00 a.m. - 5:30 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

/Keith Vicary/
Examiner, Art Unit 2183